공학석사학위논문

필기체 인식을 위한 희소 컨볼루션 신경망 모델

Sparse Convolutional Neural Network for Handwriting Recognition

2017년 8월

서울대학교 대학원

컴퓨터 공학 전공

강 우 영

필기체 인식을 위한 희소 컨볼루션 신경망 모델

Sparse Convolutional Neural Network for Handwriting Recognition

지도교수 장 병 탁

이 논문을 이학석사학위논문으로 제출함.

2017년 7월

서울대학교 대학원

컴퓨터 공학 전공

강 우 영

강우영의 석사학위논문을 인준함

2017년 7월

위 원 장 염 헌 영 (인)

부 위 원 장 장 병 탁 (인)

위 원 엄 현 상 (인)

초 록

자동화된 문자 인식기는 우편물 분류의 자동화, 번호판 인식, 전자 메모장 등 다양한 산업 분야에서 그 수요가 기하급수적으로 증가하고 있다. 이와 관련하여 최근 이미지 인식분야에서 뛰어난 성능을 보이는 컨볼루션 인공 신경망(CNN)을 사용한 방법들이 필기체 인식 분야에 적용되고 있다. 이러한 연구들 대부분에서는 높은 인식률을 달성하기 위해 주로 깊은 구조의 CNN을 사용하였다. 하지만 필기체 인식 분야에서는 주로 스마트폰이나 태블릿 PC 등 자원이 제한되어있는 단말기가 주로 사용되므로 모델이 차지하는 메모리와 계산속도 역시 중요하게 고려되어야 한다. 이에 본 논문에서는 학습 변수의 수를 효과적으로 줄이기 위해 인셉션 모듈 기반의 컨볼루션 신경망을 한글 필기체 인식문제에 적용하였다. 또한 일반화 오류를 낮추어 좀 더 높은 인식률을 달성하기 드랍필터 기법을 사용하여 컨볼루션 신경망을 희소한 성질을 가지도록 학습시켰다. 인셉션 모듈은 Imagenet Large Scale Visual Recognition Challenge 2014에서 최고의 인식률을 달성하면서도 기존의 모델에 비해 12배 적은 파라미터를 사용하여 크게 주목받은 GoogLeNet의 핵심 모듈이며, 드랍필터는 최근 널리 사용되는 regularization 기법의 일종인 드랍아웃을 CNN에 적합하게 변화를 준 기법이다. 실험은 우선 CNN에서 드랍필터의 효과를 검증하게 위해 10개 클래스, 총 60,000장의 자연 이미지로 구성된 Canadian Institute for Advanced Research(CIFAR)-10 데이터를 사용하여 드랍아웃을 적용한 모델과 인식률 비교를 수행하였다. 검증 실험을 통해 드랍필터 기법이 CNN에 적용되었을 때 드랍아웃보다 일반화 오류를 낮추는데 더 뛰어남을 확인할 수 있었다. 또한 검증 실험 중 각 은닉 층마다 드랍필터의 효과가 다르다는 것을 발견하고 이에 대

한 추가적인 검증 실험을 수행하였다. 이후 드랍필터를 인셉션 모듈에 기반하여 구성된 CNN에 적용한 뒤 한글 필기체 인식을 수행하였다. 실험에 사용한 데이터는 총 520클래스, 260,000 글자의 한글 낱글자로 이루어져 있다. 한글 필기체 인식 실험 결과 제안하는 모델인 드랍필터를 적용한 인셉션 모듈 기반의 CNN이 기존의 LeNet 구조의 CNN에 비해 3배 더 적은 학습변수로도 3.279% 높은 인식률을 달성하였다.

주요어 : 한글 필기체 인식, 컨볼루션 인공 신경망(CNN), 인셉션 모듈, 드랍필터, 드랍아웃

학 번 : 2015-22892

목 차

그림 목차

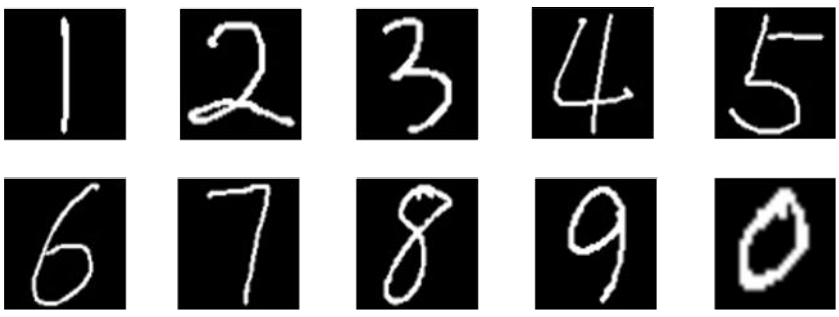
표 목차

I. 서 론

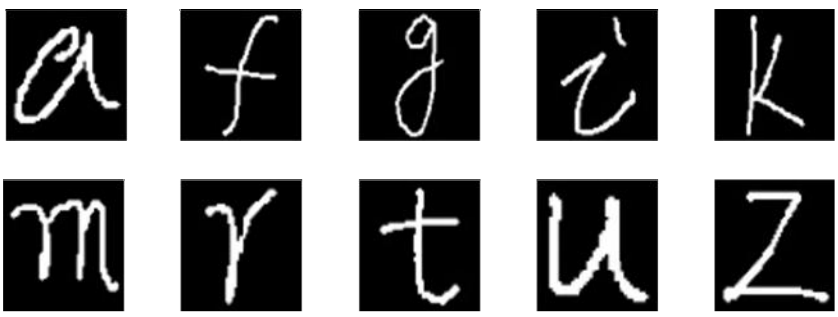
1. 연구의 필요성 및 목적

기계학습을 통한 필기체 인식에 대한 수요는 과거 자동화된 우편물의 분류, 문서의 주제별 분류 등으로 시작해서 최근 전자 단말기를 통한 손글씨의 인식까지 다양한 산업 분야에서 그 수요가 꾸준히 증가해 왔다. 따라서 과거 1990년대부터 그에 대한 활발한 연구가 시작되었으며, 당시에는 통계모델(Cho & Kim, 2003) 혹은 Support Vector Machine (SVM)을 통한 분류기가 주를 이루었다 (Bahlmann et al., 2002). 얕은 구조를 갖는 인공 신경망을 이용한 숫자 필기체 인식 연구 역시 수행된 바 있지만 (LeCun et al., 1998), 당시 학습속도, 인식률 등에 한계를 드러내며 인공 신경망을 사용한 연구는 침체되어 있었다. 하지만 최근 컴퓨터 하드웨어의 발달, 깊은 구조를 갖는 인공신경망을 효과적으로 학습하기 위한 기법들, 인식률을 올리기 위한 다양한 구조의 인공 신경망들이 등장하면서 필기체 인식 분야에서도 인공 신경망 기반 모델을 사용하는 연구의 수 역시 크게 증가하였다. 깊은 구조의 인공신경망인 딥러닝 기술을 필기체 인식에 적용한 대표적인 예 들로는 심층 컨볼루션 인공신경망 (CNN)을 이용한 방법과 순환 신경망을 이용한 방법 등이 있다 (Sermanet et al., 2012; Liwicki et al., 2007; Yin et al., 2013).

한글의 필기체 인식의 경우 역시 최근 딥러닝을 활용한 연구가 활발하게 수행되었다 (Kim & Xie, 2015; Kim et al., 2016). 한글의 경우 그림 1과 같이 숫자나 알파벳보다 더 복잡한 구조를 가진다. 또한 글자의



(a) 숫자 데이터의 예시



(b) 알파벳 데이터의 예시

(c) 한글 데이터의 예시

[그림 1] 숫자, 알파벳, 한글 데이터의 예시. 한글의 경우 숫자 (10개 클래스) 알파벳 (26개 클래스) 보다 훨씬 많은 2,000개 이상의 클래스가 존재하며 구조적으로 더 복잡하다.

종류 역시 알파벳보다 훨씬 많은 2,000자 이상이 있다. 따라서 최근 여러 이미지 인식 분야에서 뛰어난 성능을 보이는 CNN구조의 네트워크를 사용하여 한글 필기체 인식을 수행하는 것은 적절한 방법이라고 볼 수 있다. 하지만, 필기체 인식이 주로 사용되는 응용 단말기는 주로 스마트 폰이나 태블릿 PC와 같이 메모리와 GPU등 물리적 자원이 제한되는 환경이 대부분이므로 필기체 인식기를 개발함에 있어 높은 정확도와 앞서 언급한 제약 사항 등을 동시에 고려해야 한다.

본 논문에서는 제한적인 자원을 가진 환경에서 어떻게 학습 변수를 효과적으로 줄이면서도 높은 인식률을 얻을 수 있는지에 대한 연구를 첫 번째 목적으로 하였으며, 이를 달성하기 위해 인셉션 모듈(Inception module)을 사용하였다. 인셉션 모듈은 Imagenet Large Scale Visual Recognition Challenge 2014 (ILSVRC 2014)에서 ILSVRC 2012당시 최고의 성능을 보였던 Alex network(Krizhevsky et al., 2012)에 비해 12배 적은 파라미터로도 약 10%의 top-5 정확도 향상을 달성한 GoogLeNet의 핵심 모듈이다 (Szegedy et al., 2015). 이러한 장점을 취해 인셉션 모듈을 기반으로 CNN network를 구성하여 한글 필기체 인식 문제에 적용한 사례 역시 보고되었다 (Kang et al., 2016). 하지만 일반적으로 딥 러닝 모델의 경우 은닉 층을 깊게 쌓으면 쉽게 데이터를 과 적합 (overfitting) 시키는 문제가 있다. 따라서 낮은 일반화 오류(generalization error)을 달성하기 위해 적절한 regularization 기법을 사용하여 과 적합 문제를 완화시키는 것 역시 중요한 문제이다.

드랍아웃 기법은 앞서 인공신경망 기반 모델에서 언급한 과 적합 문제를 풀기위해 최근 널리 사용되고 있는 regularization 기법 중 하나이다 (Srivastava et al., 2014). 드랍아웃은 대표적인 인공 신경망인 Multi Layer Perceptron(MLP)의 은닉 층에 적용되어 임의의 확률로 각 은닉

노드들을 학습에서 배제시키는 방식으로 적용된다. 드랍아웃을 사용하면 weight decay 혹은 sparsity penalty (Ng, 2011) 같은 다른 regularizer가 없어도 학습 시 네트워크를 희소한(sparse) 성질을 가지도록 만들 수 있다 (Srivastava et al., 2014). 이를 통해 우리는 각 은닉 층의 출력으로 희소 표현(sparse representation)을 얻을 수 있으며, 이는 일반화 오류를 낮추는데 도움이 된다. 하지만 이를 CNN의 컨볼루션 층의 특징 맵에 직접 적용을 할 경우 일반화 오류를 낮추는데 유의미한 기여를 하지 않는다는 보고가 있었다 (Hinton et al., 2012). 따라서 드랍아웃 기법에 변화를 주어 CNN에 적합한 형태를 찾아 그 효과를 확인해보는 것을 두 번째 목표로 한다. 이를 해결하기 위해 드랍아웃 기법을 적용할 때 특징 맵의 임의의 픽셀 값이 아닌 임의의 마스크 필터 자체를 드랍시키는 드랍필터 기법을 사용 하였다. 이후, 이를 인셉션 모듈 기반의 CNN에 적용하여 한글 필기체에 대한 인식을 수행하였다.

실험은 우선 드랍필터의 효과에 대해 분석하기 위해 10개 클래스, 60,000장의 자연 이미지로 구성된 Canadian Institute for Advanced Research(CIFAR)-10 (Krizhevsky & Hinton, 2009)를 사용하여 이미지 분류 실험을 수행하였다. 실험을 통해 드랍필터가 CNN에 적용되었을 때 드랍아웃 보다 더 낮은 일반화 오류를 달성함을 확인하였다. 또한 드랍필터가 CNN의 첫 번째 은닉 층에 적용될 경우 오히려 인식률이 더 떨어짐을 확인할 수 있었고 이에 대한 분석 실험 역시 부가적으로 수행하였다. 이후 학습 변수의 수를 줄이기 위해 인셉션 모듈을 쌓아 만든 CNN에 드랍필터 기법을 적용하여 제안하는 모델을 구성한 후 총 520 클래스, 260,000장의 글자로 이루어진 한글 필기체 데이터에 대한 인식 실험을 수행하였다.

2. 연구 문제

본 연구에서는 필기체 인식을 물리적 자원의 제약이 있는 실제적 응용에 적용하기 위해 학습 변수의 수를 효과적으로 줄이면서도 높은 인식률을 달성할 수 있는 CNN의 구조에 대해 논한다. 이를 위해 인셉션 모듈을 사용하여 심층 CNN 모델을 구성하였다. 또한 깊은 모델에서도 일반화 오류를 낮추어 더욱 높은 인식률을 달성하기 위해 CNN에 적합한 형태의 변형 드랍아웃 기법에 대해 논한다. 구제적인 연구 문제는 다음과 같다.

첫째, 인셉션 모듈을 사용하여 CNN 모델을 만들 경우 기존의 LeNet (Lecun et al, 1998)기반의 CNN보다 적은 수의 파라미터로도 상응하는 인식률 달성이 가능한가?

둘째, 드랍필터가 드랍아웃에 비해 CNN에 적용되었을 때 더 낮은 일반화 오류를 달성할 수 있는가?

셋째, 은닉 층별로 드랍필터의 효과가 어떻게 다른가?

넷째, 드랍필터가 인셉션 모듈에 적용되어도 일반화 오류를 더욱 낮출 수 있는가?

II. 관련 연구

1. 컨볼루션 인공 신경망

1.1. 컨볼루션 연산의 정의

컨볼루션 연산은 하나의 함수에 또 다른 함수를 반전시킨 뒤 이동시키면서 곱한 다음 구간에 대해 적분하여 새로운 함수를 만드는 것이다. 두 함수 f와 g의 컨볼루션 $f*g$을 수식으로 살펴보면 식 (1)과 같다.

$$\begin{aligned}(f*g)(t) &= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau\end{aligned} \tag{1}$$

위 수식에서 t는 임의의 순간을 나타내며 τ의 경우 이동된 구간을 의미한다. 이를 컨볼루션 신경망에 적용하기 위해 이산 함수에 대해 정의하면 수식 (2)와 같다.

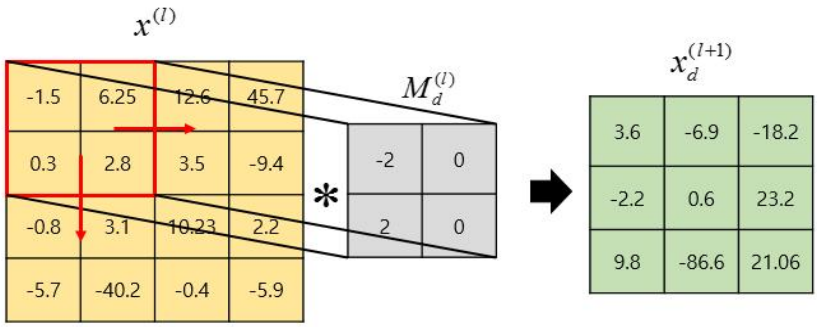
$$(f*g)(m) = \sum_{n} f(n)g(m-n) \tag{2}$$

1.2. 컨볼루션 인공 신경망

일반적인 CNN의 경우 이산적인 연산을 통해 컨볼루션을 수행하므로 식 (2)를 따른다. 이를 사용하여 CNN에서 실제 컨볼루션 연산이 어떻게 수행되는 지를 좀 더 자세히 살펴보자. l 계층에서의 특징 맵 $x^{(l)}$에 컨볼루션 연산을 취한 뒤의 결과 특징 맵 $x^{(l+1)}$이 계산되는 수식은 다음과 같다.

$$x_{i,j,d}^{(l+1)} = \sum_{w=0}^{W} \sum_{h=0}^{H} (x_{i+w,j+h}^{(l)} \times M_{w,h,d}^{(l)} + b_d^{(l)}) \tag{3}$$

식 (3)에서 i, j는 특징 맵에서 i, j번째 위치를 의미하며, W, H는 각각 l 계층의 d번째 마스크 필터 $M^{(l)}$의 너비와 높이를 나타낸다. b는 바이어스 항을 의미한다. 이를 특징 맵의 모든 위치에 대해서 수행하면 최종적인 컨볼루션 결과 특징 맵이 나온다. 이를 그림으로 개념적으로 표현하면 그림 2와 같다. 그림 2에서 $*$는 콘볼루션 연산을 의미한다.



[그림 2] 컨볼루션 연산의 예시

2. 컨볼루션 신경망을 사용한 한글 필기체 인식

최근 CNN기반의 인공 신경망들이 여러 가지 이미지 인식 분야에서 뛰어난 성능을 보이고 있다. 하지만 데이터부족, 복잡한 글자구조 등의 이유로 CNN을 활용하여 한글 필기체 인식을 수행한 연구는 비교적 많지 않다 (Kim & Xie, 2015; Kim et al., 2016). 해당 연구에서 사용한 CNN은 기본적인 LeNet 구조의 모델을 사용하였으며, 글자 인식 작업을 더 잘 수행하도록 목적 함수에 변화를 준 연구이다. 전자의 연구에서는 목적 함수로 평균 제곱 오차(Mean Squared Error)를 사용하였고, backpropagation 과정에서 발생하는 양과 음의 오류 신호 불균형을 줄이기 위해 추가적인 가중치 파라미터 α를 도입하였다. 후자의 연구에서는 비슷한 유형의 글자들을 더 잘 구분하도록 discriminative 함수를 도입하여 기존의 softmax 기반 목적 함수와 합치는 hybrid learning 알고리즘을 제안하였다. 또한 최근 필기체 인식의 실제 응용에 적합하도록 인셉션 모듈을 사용하여 한글 필기체 인식을 수행한 연구 역시 보고되었다 (Kang et al., 2016).

3. 컨볼루션 신경망의 다양한 구조

컨볼루션 신경망이 이미지 인식 분야에서 최고의 성능을 보임에 따라 공학적으로 설계된 네트워크의 중요성 역시 강조되었다 (Zhang et al, 2016). 이에 따라 CNN에 대한 다양한 구조적 연구가 수행 되었고, 그 중 최근 두드러지는 성공을 거둔 대표적인 두 가지 구조의 CNN을 소개한다.

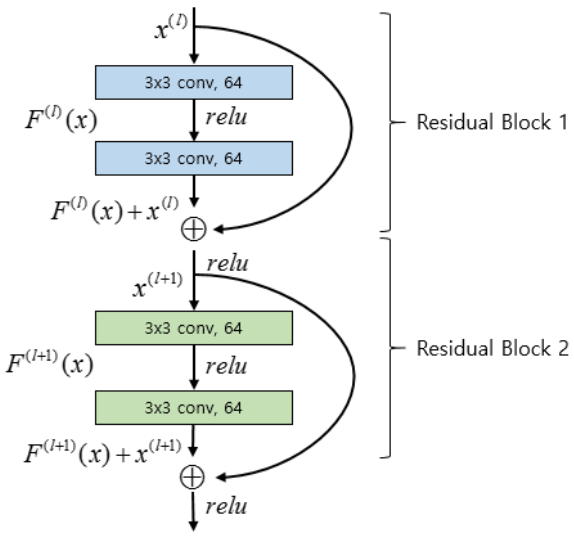
3.1. Residual Network 구조

잔차 네트워크(residual network)는 CNN의 은닉 층 사이에 지름선(shortcut connection)을 사용하여 기존의 CNN에 비해 100 층 이상으로 훨씬 깊은 네트워크를 구조화 하고 성공적으로 학습시켜 ILSVRC2015에서 최고의 성능을 보인 바 있다(He et al., 2015). 해당 연구에서는 기존의 CNN의 은닉 층이 너무 깊어지게 되면 오히려 인식률이 떨어지거나 모델의 학습이 불가능하다는 문제점을 발견하고 이를 해결하기 위해 은닉 층 사이에 지름선을 추가하였다.

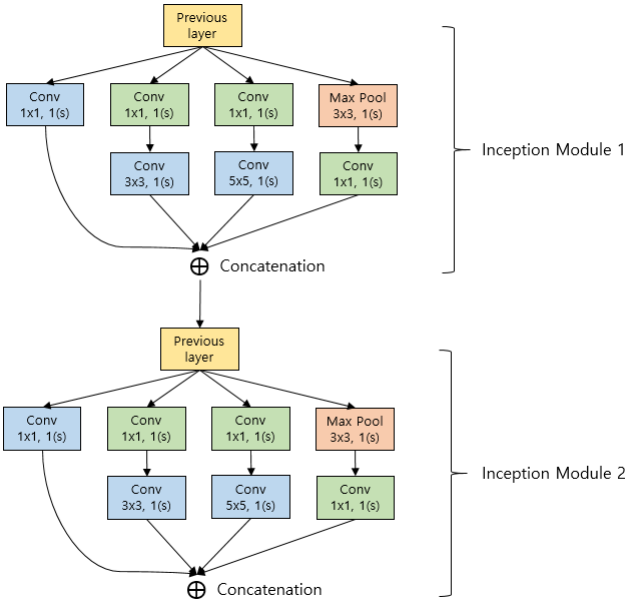
지름선이 포함된 잔차 네트워크의 학습은 매우 쉽게 구현되며 end-to-end 방식으로 학습될 수 있다. $H(x)$를 모델이 현재 은닉 층에서 본질적으로 학습하고자 하는 사상 함수라고 하자. 이때, 잔차 네트워크는 학습하고자 하는 $H(x)$를 직접적으로 학습하지 않고 이전 은닉 층의 입력인 x를 뺀 형태 즉, $H(x) - x$ 를 학습하는 것을 목표로 한다. 그 결과 x로부터 현재 은닉 층까지의 실제 비선형 사상 함수를 $F(x)$라고 했을 때, $F(x) = H(x) - x$을 학습하는 것을 목표로 하므로 $F(x) + x$형태의 새로운 비선형 사상함수가 만들어 진다. 이런 과정을 통해 잔차 네트워크는 비 잔차 네트워크의 성능을 최대한 유지시키면서 x를 현재 은닉 층의 또 다른 참조특징(reference feature)로 사용하여 더욱 좋은 성능을 낼 수 있게 된다. 지름선이 연결된 잔차 네트워크의 일부에 대한 개념도가 그림 3에 나타나 있다.

3.2. GoogLeNet 구조

GoogLeNet은 인셉션 모듈을 사용하여 깊은 구조의 CNN을 만든 후 ILSVRC2014에서 당시 뛰어난 분류 성능을 자랑하는 Alex network (Krizhevsky et al., 2012)에 비해 12배 적은 학습 변수로도 최고의 이미지 인식률을 달성하였다 (Szegedy et al., 2015). 이러한 성능을 달성하기 위해 GoogLeNet에서는 인셉션 모듈을 사용하였다. 인셉션 모듈이 가지는 장점에 기여하는 요인을 살펴보면 크게 두 가지를 둘 수 있다. 먼저 학습 변수의 양을 효과적으로 줄일 수 있는 1x1 크기의 축소 마스크(reduction mask)가 있다. 이는 두 은닉 층 사이에 또 다른 작은 인공 신경망을 두어 컨볼루션 마스크가 좀 더 복잡하고 구별적인 특징을 잘 찾아낼 수 있게 고안된 모델인 network in network (Lin et al., 2013)에서 영감을 얻은 방법이다. GoogLeNet에서는 작은 인공 신경망을 1x1 크기의 컨볼루션 마스크로 대치하여 입력 특징 맵의 채널을 효과적으로 줄였고 이는 결국 전체 학습 파라미터를 감소시키는 역할을 하였다. 두 번째로는 다양한 크기의 컨볼루션 마스크(convolutional mask)들이 있다. 1x1, 3x3, 5x5 등 다양한 크기를 가지는 컨볼루션 마스크를 concatenation시켜서 다양한 공간적 크기를 가지는 패턴들을 효과적으로 학습할 수 있게 하였다. 이에 대한 개념도가 그림 3에 나타나 있다. 더 가벼운 모델로도 더 높은 인식률을 달성할 수 있다는 장점은 실제적인 필기체 인식 응용에 적합한 면을 보여서, 본 논문에서 제안하는 모델의 기본 구조로 사용하였다.



(a) Residual Network



(b) GoogLeNet

[그림 3] Residual Network와 GoogLeNet의 일부분에 대한 개념도

4. 인공 신경망에서의 Regularization

4.1. 다층 퍼셉트론에서의 Regularization

인공 신경망 기반 모델이 깊어지고 학습변수 역시 기하급수적으로 늘어남에 따라 과 적합 문제를 완화시키기 위해 regularization 기법의 일종으로 드랍아웃 기법이 제안되었다 (Hinton et al., 2012). 드랍아웃 기법은 임의의 확률 p로 은닉 층의 임의의 노드를 값을 0으로 변경하여 학습에 참여시키지 않는 방법이다. 이는 실제 학습과정에 관여하는 은닉 노드들의 수를 감소시킴으로 인해 과 적합 문제를 완화시킬 수 있는 방법이다. 또 다른 중요한 해석은 드랍아웃이 임의로 선택된 서브 네트워크들의 앙상블 학습을 야기하며 인식률을 크게 향상시키는 데 기여한다는 것이다. 이와 비슷한 방법으로는 드랍커넥트 (dropconnect)라는 기법이 있다(Wan et al., 2013). 드랍커넥트는 드랍아웃을 일반화 시킨 개념으로 완전 연결된(fully connected) 다층 퍼셉트론 구조의 인공 신경망에서 임의의 가중치 변수 값을 0으로 변경하는 방법이다. 이를 통해 네트워크는 전체 가중치 변수의 부분집합들로 학습되며 이 역시 모델의 과 적합을 완화해 준다.

4.2. 컨볼루션 인공 신경망에서의 Regularization

컨볼루션 신경망에서의 regularization 기법 역시 다양하게 연구되었다. stochastic pooling의 경우 기존에 deterministic한 방법으로 pooling 연산을 수행하던 것을 확률적인 연산을 통해 pooling 시키는 방법을 취한다 (Zeiler & Fergus, 2013). 다른 기법으로는 maxout network가 있는데, 각 은닉 층의 특징 맵에서 가장 큰 활성화 값을 가지는 노드 값만을 선택적으로 출력하는 방법으로 드랍아웃 기법과 결합되어 인상적인

regularization 효과를 보인바 있다 (Goodfellow et al,. 2013). 또한 (Gal & Ghahramani, 2015)에서는 bayesian convolutional neural network를 제안하였다. 해당 논문에서는 드랍아웃이 적용된 인공신경망을 베이지안 인공신경망의 일종으로 유도하여 Monte-Carlo 드랍아웃 기법을 통해 모델을 학습시켰다. (Tompson et al., 2015)에서는 각 은닉 층의 출력 특징 맵에서 임의의 채널을 골라 채널단위로 드랍아웃을 하는 spatial dropout 기법을 제안하기도 하였다.

III. 제안하는 모델

1. 컨볼루션 신경망에서의 드랍아웃

드랍아웃은 일반적으로 완전 연결된 은닉 층(fully-connceted layer)에서 주로 적용되어 일반화 오류를 낮추는데 크게 기여하였다. 컨볼루션 신경망에서 드랍아웃이 어떻게 적용되는지 살펴보기에 앞서 다층 퍼셉트론(MLP)에서 드랍아웃이 학습 과정에서 어떻게 적용되는지 수식을 통해 먼저 살펴보자. 수식을 살펴보기에 앞서 각 변수들에 대한 표기법을 정의하겠다. 먼저 L개의 은닉 층을 가지는 인공 신경망을 생각해보자. 이 때, 은닉 층의 인덱스를 $l \in \{1,...,L\}$와 같이 표현할 수 있다. 또, $z^{(l)}$을 l 층의 입력 벡터로, $o^{(l)}$을 l 층의 출력 벡터로 정의한다. $W^{(l)}, b^{(l)}$을 각각 l 층의 가중치와 편향(bias) 벡터로 표현한다. 일반적인 MLP구조에서 전방 전파(feed-forward)과정을 표현하면 식 (4)와 같다.

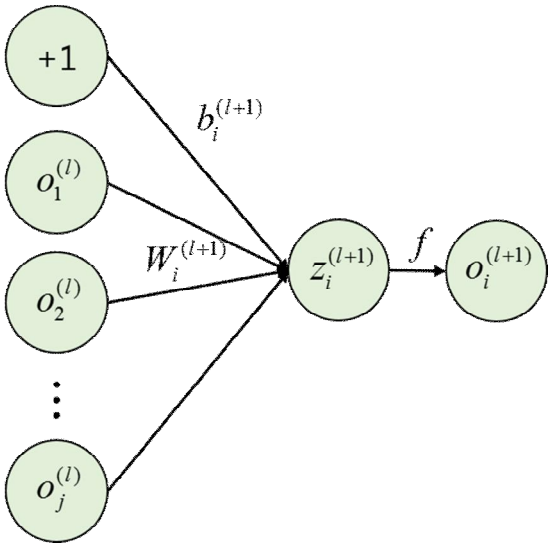
$$\begin{aligned} z_i^{(l+1)} &= W_i^{(l+1)} o^l + b_i^{(l+1)}, \\ o_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned} \tag{4}$$

식 (4)에서 $f(\bullet)$은 비 선형 활성화 함수를 의미하며 예를 들면, $f(x) = 1/(1+\exp(-x))$ 와 같이 시그모이드 함수가 있다. 또한 i는 $(l+1)$ 층에서 은닉 뉴런의 인덱스를 의미한다. 이후 드랍아웃이 적용되면 전방 전파에 대한 수식은 식 (5)와 같이 바뀐다.

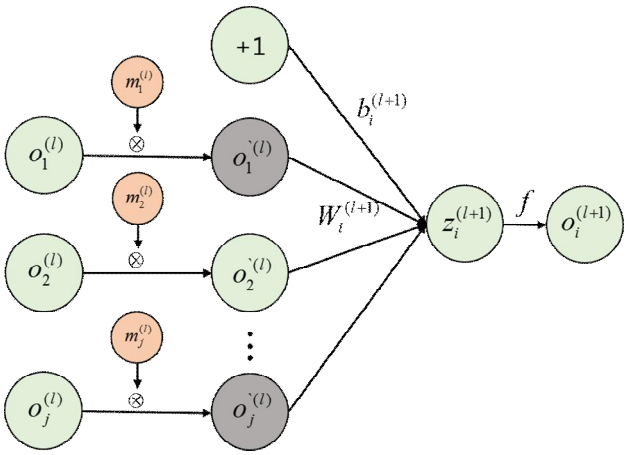
$$
\begin{aligned}
m_j^{(l)} &\sim Bernoulli(p), \\
o^{'(l)} &= m^{(l)} \otimes o^{(l)}, \\
z_i^{(l+1)} &= W_i^{(l+1)} o^{'(l)} + b_i^{(l+1)}, \\
o_i^{(l+1)} &= f(z_i^{(l+1)}),
\end{aligned}
\tag{5}
$$

식 (5)에서 $\otimes$은 요소 곱(element-wise product)을, j는 l 층 은닉 뉴런의 인덱스를 의미한다. 또 임의의 은닉 층 l에 대하여, $m^{(l)} \in \{0,1\}^j$은 임의의 확률 p로 1의 값을 가지는 베르누이 확률 변수 벡터 이다. 이는 0 또는 1의 값을 가지며 크기가 $o^{(l)}$과 동일하므로 드랍아웃 마스크라고 부를 수 있다. 이를 통해 전체 네트워크는 학습 시 임의의 은닉 뉴런 값이 0으로 변경되므로 전체 네트워크에 대한 서브 네트워크로 여겨질 수 있다. 주의해야 할 점은, 학습 종료 후 실제 인식을 수행할 때에는 $W^{(l)}$에 드랍아웃 확률 p를 곱해주어야 한다는 것이다 (Hinton et al., 2012). 이를 그림으로 살펴보면 (그림 4)과 같다.

CNN에서의 드랍아웃은 MLP에서의 드랍아웃과 전체적으로 비슷한 과정을 따른다. 차이점은 MLP의 경우 하나의 입력 객체에 대한 은닉 층의 출력은 $o_{mlp}^{(l)} \in R^{j \times 1}$ 크기의 벡터가 되지만, CNN의 경우 은닉 층에서의 출력은 $o_{cnn}^{(l)} \in R^{h \times w \times d}$ 크기의 tensor 형태를 가진다. 이때, h, w, d는 각각 출력 특징 맵의 높이, 너비, 채널을 의미한다. 따라서 특징 맵에 원소 곱이 수행되는 드랍아웃 마스크의 크기 역시 $m_{cnn}^{(l)} \in \{0,1\}^{h \times w \times d}$이 된다. tensor 형태의 드랍아웃 마스크가 특징 맵에 요소 곱이 되면 이후 드랍아웃된 특징 맵은 컨볼루션 마스크에 의해 컨볼루션 연산이 수행되며 이후 과정은 일반적인 CNN과 동일하다. 이를 드랍필터와의 차이에 대해 개념적으로 설명한 그림은 다음 절에서 다루도록 하겠다.



(a) 드랍아웃이 적용되지 않은 다층 퍼셉트론 구조의 신경망



(b) 드랍아웃이 적용된 다층 퍼셉트론 구조의 신경망. 회색으로 음영된 은닉 뉴런은 드랍아웃에 의해 값이 0으로 변경된 뉴런을 의미한다.

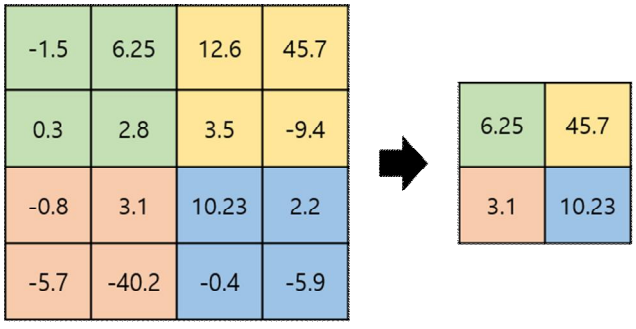
[그림 4] 드랍아웃이 적용되었을 때 네트워크의 차이

2. 컨볼루션 신경망에서의 드랍필터

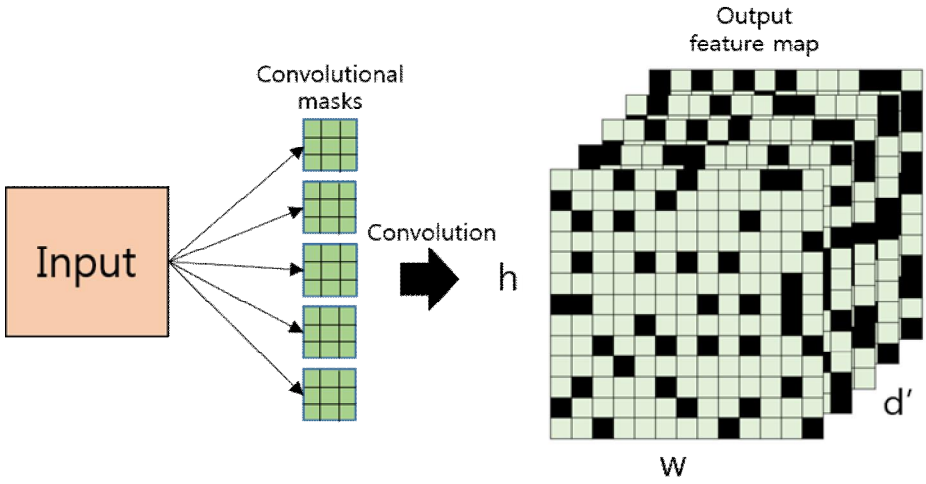
(Tompson et al., 2015)에서는 CNN 모델 각 은닉 층의 출력인 특징 맵들에 대해 픽셀 단위로 드랍아웃을 적용하는 방법이 과 적합을 효과적으로 완화시키지 못한다고 주장하였다. 이를 해결하기 위해 그들은 자연 이미지의 경우 강력한 공간적 연관성(spatial correlation)이 있으며, 그것이 컨볼루션 레이어를 거치면서 비 선형적으로 변화한다고 해도 여전히 특징 맵에서는 강한 공간적 연관성이 있다고 가정하였다. 따라서 각각의 픽셀 하나하나를 드랍시키는 방법 대신 드랍되는 픽셀의 인접 픽셀들 역시 같이 드랍시키는 spatial dropout을 제안하였다. 이를 조금 더 확장하면 인접 픽셀의 범위를 특징 맵의 특정 채널 전체로 생각하고 드랍시킬 수 있다. 본 논문에서 사용하고자 하는 드랍필터의 경우 임의의 마스크 필터 값을 전부 0으로 변경하면 컨볼루션 결과 생성되는 특징 맵에서 드랍된 마스크 필터에 대응하는 채널 값이 전부 0이 된다. 이 방법은 계산적 결과로서 spatial dropout과 같아지게 되지만, 우리는 CNN의 앙상블 학습 효과에 초점을 두어 각 컨볼루션 마스크 필터를 드랍시키는 의미에서 이 기법을 드랍필터라고 명명 지었다. 임의의 확률 p로 CNN의 임의의 컨볼루션 마스크를 드랍시키는 방법에 대해 수식으로 살펴보면 식 (6)와 같다.

$$
\begin{aligned}
m^{(l)} &\sim Bernoulli(p), \\
W^{'(l)} &= m^{(l)} \otimes W^{(l)}, \\
z_i^{(l+1)} &= W_i^{(l+1)} * o^{(l)} + b_i^{(l+1)}, \\
o_i^{(l+1)} &= P(f(z_i^{(l+1)})),
\end{aligned}
\tag{6}
$$

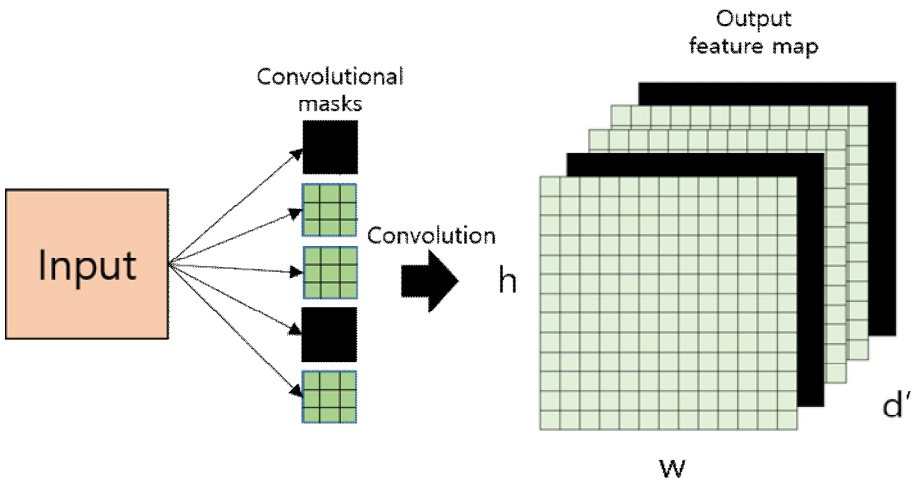
$m^{(l)} \in \{0,1\}^{d'}$은 드랍아웃 마스크를 나타내고 d'는 l 층에서의 컨볼루션 마스크 필터의 개수를 의미하며 이는 $(l+1)$층에서의 특징 맵의 채널과 동일하다. $W^{(l)} \in R^{h' \times w' \times d \times d'}$은 컨볼루션 마스크이며, h', w', d는 각각 마스크 필터의 높이, 너비, 채널을 의미한다. 이때 마스크 필터의 채널 d는 l 층에서 특징 맵의 채널 수와 동일하다. 즉, 드랍아웃 마스크를 현재 은닉 층의 컨볼루션 마스크와 채널 단위 곱셈 $\otimes$을 수행하여 실제 학습에 참여할 임의 개수의 마스크 필터를 선택한다. 이후 선택된 컨볼루션 마스크의 부분집합만을 가지고 컨볼루션 연산 $*$을 수행한 뒤, 비선형 함수 $f(\bullet)$와 최댓값 풀링(max pooling) $P(\bullet)$을 거친다. 최댓값 풀링 연산의 경우 $n \times n$ 크기의 풀링 마스크를 만든 뒤, s 크기의 stride로 마스크를 이동시키면서 마스크 내의 최댓값을 최종 출력으로 만드는 방법이다. 이를 그림으로 표현하면 그림 5와 같다. 드랍 필터를 적용하면 학습 시 학습에 참여하는 컨볼루션 마스크의 부분 집합이 드랍필터에 의해 매번 바뀌므로 여러 서브 네트워크에 의한 앙상블 학습으로 볼 수 있다. 제안하는 드랍필터 기법을 드랍아웃 기법과의 차이에 대해 그림 6에서 묘사하였다.



[그림 5] 최댓값 풀링. 사용된 풀링 마스크의 크기는 2×2이며 stride역시 2로 하였다.



(a) CNN에서의 드랍아웃

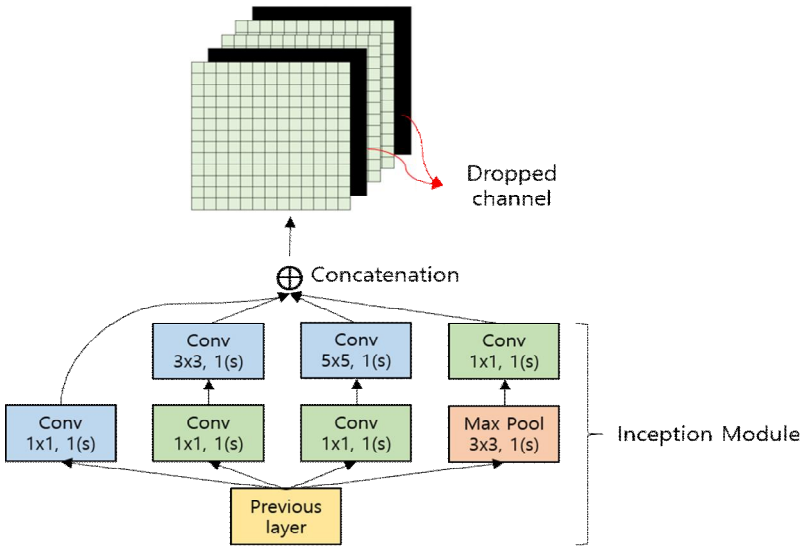


(b) CNN에서의 드랍필터

[그림 6] 드랍아웃과 드랍필터의 차이 개념도. 특징 맵에서 검은색으로 칠해진 영역이 드랍된 영역이다.

3. 드랍필터가 적용된 인셉션 모듈

인셉션 모듈은 최근 학습 파라미터의 감소를 통해 계산시간과 메모리 요구를 줄여줌과 동시에 뛰어난 인식 성능으로 널리 사용되고 있는 GoogLeNet의 핵심 구성 요소이다. 인셉션 모듈은 기본적인 CNN의 컨볼루션 연산에 비해 복잡한 구조를 가지므로 드랍필터를 적용시키는 방법에 대해서도 생각해 보아야 한다. 자세한 분석 및 응용은 추후 연구해 볼 것이며, 본 논문에서는 드랍필터의 regularization 효과에 주목하기 위해 휴리스틱한 방법을 취하였다. 우리는 3×3 또는 5×5 컨볼루션 마스크 전에 수행되는 채널 축소(depth reduction) 마스크에 대해 드랍필터를 사용하지 않았다. 그 이유는 채널 축소의 경우 특징 맵의 채널을 효율적으로 줄이는 역할을 하므로 특정 채널 축소 마스크가 드랍될 경우 손실되는 정보가 매우 클 것으로 가정했기 때문이다. 드랍필터가 적용된 인셉션 모듈에 대한 개념도를 그림 7로 묘사하였다.



[그림 7] 드랍필터가 적용된 인셉션 모듈

IV. 실험 및 필기체 인식 결과 분석

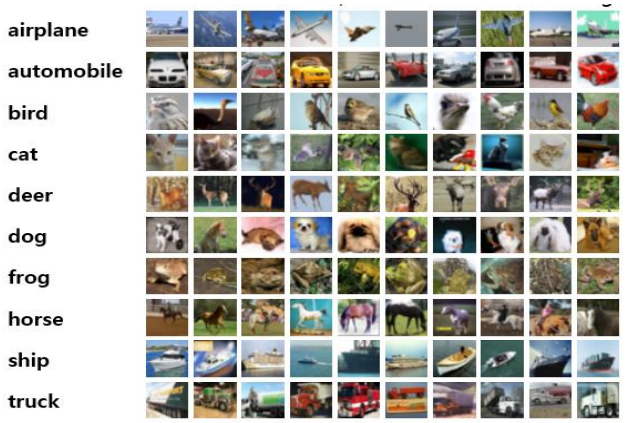
1. 데이터 명세

실험에 사용된 데이터는 총 2가지가 있다. 첫째로는 먼저 드랍필터의 효과에 대해 빠르게 검증하기 위해 사용된 CIFAR-10 (Krizhevsky & Hinton, 2009) 데이터가 있다. CIFAR-10 데이터는 총 10개 클래스, 60,000장의 자연 컬러 이미지로 구성되어 있다. 그림 8에서 CIFAR-10 데이터에 대한 예를 보여준다.

두 번째로는 한글 필기체 데이터이다. 실험을 위해 다양한 연령, 직업, 성별로부터 한글 필기체 데이터를 직접 수집하였다. 데이터에 대한 예시는 그림 1에서 살펴 보았다. 그림 1처럼 한글 필기체는 간단한 글자부터 복잡한 구조까지 매우 다양한 분포를 가진다. 또한 비슷하게 생긴 글자 역시 많은 편이다. 따라서 한글 필기체에 대한 인식은 필기체 인식 문제 중에서도 어려운 편에 속한다고 볼 수 있다. 데이터 집합은 총 520개 클래스이며 각 클래스마다 500개의 샘플로 이루어져 총 260,000자의 글자로 구성되어 있다. 이러한 구성은 1990년대에 수집되어 최근 (Kim & Xie, 2015; Kim et al., 2016)에서 사용된 SERI95a 데이터 집합과 동일한 클래스 구성을 따른다. 수집된 한글 필기체 데이터 집합에 포함된 모든 클래스는 표 1에 명시되어 있다.

[표 1] 한글 필기체 데이터의 총 클래스. 520개 클래스가 있다.

가	곳	김	놀	독	뛰	록	맹	반	비	수	암	열	월	쟁	증	청	코	퍼	헌
각	공	깊	놈	돈	뜻	론	머	받	빈	숙	압	염	위	저	지	체	크	편	험
간	과	까	농	돌	라	롯	먹	발	빛	순	았	였	유	적	직	쳐	큰	평	혀
갈	관	깨	높	동	락	뢰	먼	밝	빠	술	앙	영	육	전	진	초	큼	폐	혁
감	광	께	놓	되	란	료	메	방	빨	스	앞	예	율	절	질	촉	키	포	현
갔	괴	꾸	누	된	랄	루	며	배	뿐	슨	애	오	으	젊	짐	촌	킬	폭	협
강	교	끌	눈	될	람	룩	면	백	사	슬	액	온	은	점	집	총	타	표	형
갖	구	끝	느	두	랍	류	명	버	산	습	야	올	을	접	짓	최	탄	푸	호
같	국	끼	는	둔	랑	르	몇	번	살	승	약	옹	음	정	징	추	탈	풀	혹
개	군	나	늘	둘	래	른	모	벌	삼	시	양	와	읍	제	째	축	태	품	혼
거	굳	난	능	뒤	략	를	목	범	상	식	어	완	응	져	쪽	출	택	풍	홍
건	굴	날	니	드	량	름	몰	법	새	신	억	왔	의	졌	차	충	터	프	화
걸	궁	남	닌	득	러	릇	몸	베	색	실	언	왕	이	조	착	취	테	피	확
검	권	납	님	든	런	리	못	변	생	심	얻	왜	익	족	찬	측	토	필	환
것	귀	났	다	들	럼	린	무	별	서	십	얼	외	인	존	찰	층	통	하	활
게	규	내	단	듯	럽	릴	문	병	석	싶	엄	요	일	좀	참	치	퇴	학	황
겠	균	낸	달	등	렇	림	물	보	선	싸	업	욕	임	종	창	칙	투	한	회
겨	그	너	담	디	레	립	므	복	설	써	없	용	입	좋	찾	친	트	할	획
격	극	넓	답	따	려	마	미	본	섭	쓰	엇	우	있	죄	채	칠	특	함	효
견	근	넘	당	땅	력	막	민	볼	성	씨	었	욱	자	주	책	침	튼	합	후
결	글	네	대	때	련	만	밀	봉	세	아	에	운	작	죽	처	칭	틀	항	훈
경	금	녀	더	떠	렬	많	및	부	셨	악	엔	울	잔	준	척	카	티	해	휘
계	급	년	덕	떤	렸	말	밑	북	소	안	엘	움	잘	줄	천	캄	파	했	흥
고	기	념	던	떨	령	망	바	분	속	앉	여	웅	잡	중	철	커	판	행	희
곡	긴	노	데	떻	례	맞	박	불	손	않	역	워	장	즈	첩	케	팔	향	히
골	길	논	도	또	로	매	밖	브	송	알	연	원	재	즉	첫	켜	패	허	힘



[그림 8] CIFAR-10 데이터의 예시

(그림 출처: https://www.cs.toronto.edu/~kriz/cifar.html)

2. 드랍필터의 효과 분석

드랍필터를 인셉션 모듈 기반 CNN에 직접 적용하여 한글 필기체 인식을 수행하기 전에 드랍필터가 CNN에 적용되었을 때 드랍아웃에 비해 일반화 오류를 더 낮출 수 있는지, 어떤 식으로 적용해야 최고의 성능을 낼 수 있는지에 대해 분석해 볼 필요가 있다. 두 실험에 대해 사용한 데이터는 CIFAR-10 데이터 집합을 사용하였다. 먼저 드랍필터를 드랍아웃과 비교하는 실험을 우선 수행하였다. 실험에 사용한 네트워크의 구조는 4층의 컨볼루션 계층과 2층의 완전 연결 층으로 구성되어 있다. 각 컨볼루션 계층의 마스크 필터 수는 각 64, 128, 256, 512개이며, 마스크 필터의 높이와 너비는 각 4x4, 4x4, 5x5, 5x5로 하였다. 완전 연결 층에는 각 512개 10개의 은닉 뉴런으로 구성되어 있다. 실험에서 비선형 활성화 함수로는 ReLU(Nair & Hinton, 2010)를 사용하였으며 학습의 가속화를 위해 모든 컨볼루션 계층에 Batch Normalization기법 (Ioffe & Szegedy, 2015)을 사용하였다. 비교를 위해 컨볼루션 계층들에 드랍아웃과 드랍필터를 0.5의 드랍 확률로 각각에 적용한 두 개의 모델을 만들어 200 epoch동안 학습한 뒤 비교 실험을 수행하였다 (표 2). 표 2에서 No drop은 드랍아웃이나 드랍필터가 적용되지 않은 모델을 의미한다. 실험 결과 드랍아웃을 사용할 경우 드랍아웃이나 드랍필터를 적용하지 않은 모델보다 인식률이 오히려 낮아짐을 확인할 수 있었다. 반면 드랍필터를 적용한 모델은 드랍필터를 적용하지 않은 모델에 비해 4.36% 높은 테스트셋 인식률을 보였다. 이를 통해 CNN에 드랍필터를 적용하였을 때 드랍아웃에 비해 일반화 오류를 더 잘 낮출 수 있음을 확인하였다.

이후 수행한 실험은 컨볼루션 계층마다 드랍필터 기법을 개별적으로 적용해보는 실험이었다. 이는 CNN의 깊은 이해에 대한 주제를 다룬 연

구로부터 동기를 얻었다. 컨볼루션 레이어를 깊이 쌓은 심층 CNN에서는 각 은닉 층마다 특징 맵에 대한 추상화 정도가 다르며 독특한 특징을 가지고 있다 (Zeiler & Fergus, 2014). 이때, 낮은 단계의 은닉 층에서는 간단한 직선이나 곡선과 같은 기본적인 패턴들에 반응하도록 컨볼루션 마스크가 학습된다. 반면, 높은 단계의 은닉 층에서는 사람의 얼굴, 자동차의 바퀴 등 클래스의 특정 부분에 상응하는 높은 추상화 정도를 보이며 해당 패턴들을 찾아낼 수 있도록 컨볼루션 마스크가 학습된다. 이러한 특성을 고려하여 우리는 드랍필터가 낮은 층에 적용되었을 때 일반화 오류를 낮추는 데 기여가 크지 않을 것이라고 가정하였다. 이를 정리하면 다음과 같다.

- CNN 모델의 낮은 층의 마스크 필터들은 직선이나 곡선과 같은 기본적인 패턴들에 반응하는 역할을 하므로 모든 클래스에 걸쳐 일반적인 패턴 조합들을 찾아준다. 따라서 과 적합의 요인이라고 볼 수 없다.

위의 가정을 검증하기 위해 본 논문에서는 드랍필터 기법을 우리가 구성한 CNN 모델에 대하여 첫 번째 층부터 마지막 층까지 각각 개별적으로 적용하고, 적용된 층에 따라 일반화 오류에 유의미한 차이가 있는지 확인하는 실험을 하였다. 검증 실험에 사용한 모델은 총 4계층의 컨볼루션 레이어로 구성되어 있으며, 이후 두 완전 연결 층으로 연결된다. 각 은닉 층마다 128 개의 컨볼루션 마스크가 존재하며 컨볼루션의 stride는 1로 동일하다. 또한 마스크의 사이즈 역시 모두 3×3으로 동일하게 맞춰주었다. 이는 각 층별 드랍필터의 효과를 정확하게 측정하기 위하여 다른 변인들은 모두 통제하고자 했기 때문이다. 이어지는 두 완전 연결 층의 은닉 뉴런의 수는 각각 512, 10으로 구성 하였다. 사용된 드랍필터의

드랍 확률은 역시 모두 0.5이며, 비선형 활성화 함수는 ReLU를 사용하였다. 또한 학습의 가속화를 위해 모든 컨볼루션 계층에 Batch Normalization기법을 사용하였다. 학습은 총 200 epoch을 수행하였으며, 실험 결과는 표 3와 같다. 표 3에서 Conv1~4의 경우 각 층에 개별적으로 드랍필터가 적용되었음을 의미한다. 실험 결과 첫 번째 컨볼루션 계층에서 드랍필터가 적용되었을 때 일반화 오류가 감소하지 않는 것을 확인할 수 있었다. 반면, 이후의 모든 층에서는 드랍필터가 적용되었을 때 일반화 오류가 감소되는 것을 확인할 수 있었다. 주목할 점은 드랍필터가 적용되었을 때 가장 효과적인 계층은 세 번째 계층인 것으로 확인되었으며, 이에 대한 검증을 위해 추가적인 실험을 수행하였다. 추가적인 실험에서 사용한 모델은 5층의 컨볼루션 계층이 있고, 이후 동일하게 두 층의 완전 연결 계층이 따라오는 형태이다. 컨볼루션 계층의 수를 제외한 모든 변인은 동일하게 설정하였다. 이에 대한 실험 결과는 표 4와 같다. 흥미롭게도 이 역시 3번째 계층에서 드랍필터의 효과가 가장 크게 나타났으며 첫 번째 층에 드랍필터를 적용시킬 경우 일반화오류를 낮추는데 도움이 되지 않았다. 이를 통해 앞서 언급한 첫 번째 은닉 층에 드랍필터를 적용하지 않는 것에 대한 가정이 맞음을 확인하였다. 추가적으로 드랍필터의 드랍 확률이 인식률에 미치는 영향에 대한 실험 역시 수행하였다 (그림 9). 이는 기존에 알려진 드랍아웃에서의 경우 0.4와 0.6 사이일 때 가장 좋은 인식률을 얻는다고 알려져 있으며 이러한 사실이 드랍필터에도 적용되는지 대한 실험이다. 실험 결과 기존의 드랍아웃에서와 마찬가지로 0.4에서 0.6 사이일 때 테스트 셋에 대한 인식률이 가장 좋게 나타났다. 앞의 실험 결과들을 토대로 가장 적절한 실험 설정을 찾아 그에 대한 결과를 표 5에 제시하였다.

[표 2] 드랍아웃과 드랍필터의 성능비교

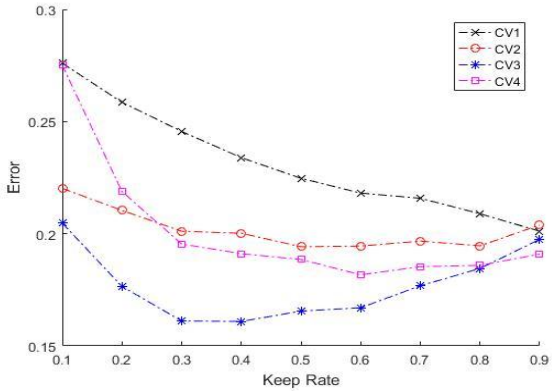
Model	Accuracy		
	Training	Validation	Test
No drop	0.9989	0.8038	0.8015
Dropout	0.9107	0.776	0.7654
Dropfilter	0.9886	0.8532	0.8452

[표 3] 4개의 컨볼루션 레이어에 각각 드랍필터를 개별적으로 적용하였을 때의 인식 정확도

Model	Accuracy		
	Training	Validation	Test
No drop	1.0	0.8182	0.8061
Conv1	0.9998	0.7984	0.7907
Conv2	0.9998	0.8221	0.8189
Conv3	0.99987	0.8397	0.8364
Conv4	0.9996	0.8273	0.82

[표 4] 5개의 컨볼루션 레이어에 각각 드랍필터를 개별적으로 적용하였을 때의 인식 정확도

Model	Accuracy		
	Training	Validation	Test
No drop	0.99915	0.8108	0.8035
Conv1	0.9992	0.7916	0.7831
Conv2	0.9997	0.8119	0.807
Conv3	0.9998	0.8457	0.8409
Conv4	0.99947	0.8171	0.8112
Conv5	0.9992	0.8217	0.8185



[그림 9] 드랍필터에서 드랍 확률별 테스트 셋에 대한 에러

[표 5] 여러 계층에 드랍필터를 적용했을 때의 성능

Model	Accuracy		
	Training	Validation	Test
No drop	0.9998	0.8125	0.8052
All dropout	0.998	0.8467	0.8351
Drop234	0.9992	0.8671	0.8613
Drop234 (Fine-Tuning)	0.99873	0.8658	0.86

표 5에서 All drop은 모든 컨볼루션 층에 드랍필터를 적용한 것을 의미하고, Drop234는 첫 번째 층을 제외하고 적용한 모델을 의미한다. Fine-Tuning의 경우 그림 9에서 보듯 각 컨볼루션 계층마다 최고의 성능을 보였던 0.9, 0.5, 0.4, 0.6을 각 층의 드랍확률로 적용시킨 모델이다. 실험 결과 드랍필터를 첫 번째 층에 적용시키지 않았을 때 더 좋은 인식률을 보였다. 또한 드랍 확률을 0.5로 한 경우와 fine-tuning한 경우 성능에 있어서 큰 차이를 보이지 않았는데, 이는 드랍 확률을 0.4~0.6 사이의 값으로 적용하면 좋은 regularization 성능을 낼 수 있음을 의미한다.

3. 필기체 인식 결과 및 분석

이전 실험 결과들을 통해 드랍필터가 CNN에 적용되었을 때 유의미한 regularizer 역할을 할 수 있으며, 특히 첫 번째 컨볼루션 계층에서는 적용시키지 않는 것이 좋다는 것을 확인하였다. 이를 바탕으로 인셉션 모듈에 기반한 CNN에 드랍필터를 적용하여 한글 필기체에 대한 인식 실험을 수행하였다. 실험에서는 역시 드랍필터의 효과에 주목하기 위해 최근 흔히 수행하는 데이터 증강(data augmentation)기법을 사용하지 않았다. 제안하는 모델의 비교 실험 모델로는 LeNet 기반의 CNN을 사용하였으며, 드랍필터의 적용 유무를 번갈아가며 실험을 수행하였다. 이후 제안하는 모델인 인셉션 모듈 기반 CNN 역시 드랍필터의 적용 유무를 바꾸어서 동일한 데이터에 대해 실험을 수행하여 인식 성능을 비교하였다. 실험에 사용한 LeNet기반의 CNN과 인셉션 모듈 기반 CNN에 대한 구조를 표 6과 표 7에 각각 나타내었다.

[표 6] LeNet 기반 CNN의 구조

Layers	Size of filters / stride	# of parameters
Conv 1	5x5x64 + 64 /1	1664
Pool 1	2x2 / 2	-
Conv 2	5x5x64x128 + 128 / 1	204,948
Pool 2	2x2 / 2	-
Conv 3	4x4x128x256 + 256 / 1	524,544
Pool 3	2x2 / 2	-
Conv 4	4x4x256x512 + 512 / 1	2,097,664
Pool 4	2x2 / 1	-
FC 1	512x384 + 384	196,992
FC 2	384x520 + 520	200,200
Total	-	3,226,012

[표 7] 인셉션 모듈 기반 CNN의 구조

Inception layer 1		
1x1	3x3 reduction	3x3
1x32	1x48 + 48	9x48x32
5x5 reduce	5x5	Pool projection
1x16 + 16	25x16x16	1x32
Output size		# of parameters
30x30x112		20,416

Inception layer 2		
1x1	3x3 reduction	3x3
1x112x64	1x112x64x + 64	9x64x64
5x5 reduce	5x5	Pool projection
1x112x16 + 16	25x16x48	1x112x64
Output size		# of parameters
15x15x240		79,440

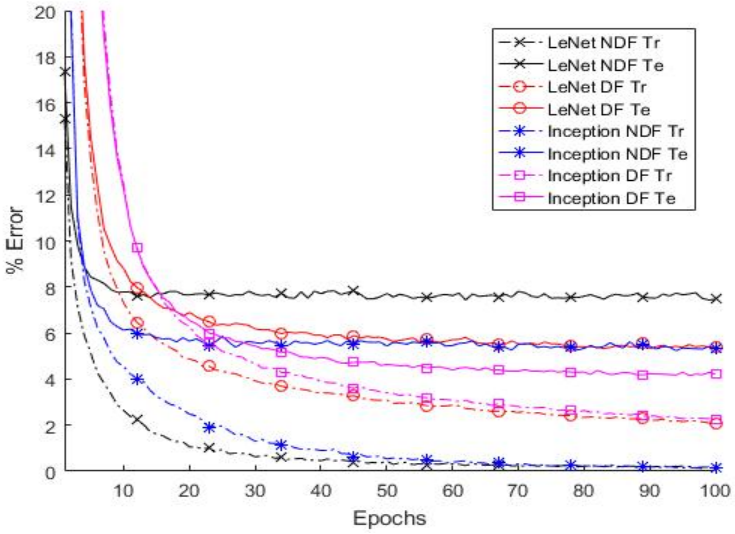
Inception layer 3		
1x1	3x3 reduction	3x3
1x120x64	1x240x64 + 64	9x64x128
5x5 reduce	5x5	Pool projection
1x240x32 + 32	25x32x128	1x240x128
Output size		# of parameters
8x8x448		245,343

Inception layer 4		
1x1	3x3 reduction	3x3
1x448x128	1x448x96 + 96	9x96x128
5x5 reduce	5x5	Pool projection
1x448x48 + 48	25x48x128	1x448x128
Output size		# of parameters
1x1x512		443,536

FC Layers	Output size	# of parameters
FC 1	512x384 + 384	196,992
FC 2	384x520 + 520	200,200
Total	-	1,185,927

인셉션 모듈 기반의 CNN이 LeNet 기반의 CNN보다 적은 학습변수의 수로도 더 높은 인식률을 달성한다는 것을 증명하기 위해 제안하는 모델을 LeNet 기반의 CNN에 비해 3배적은 수의 학습변수로 학습시켰다. 드랍필터의 경우 앞 절의 실험 결과를 토대로 첫 번째 컨볼루션 계층을 제외한 모든 컨볼루션 계층에 0.5의 드랍률로 적용시켰으며, 첫 번째 완전 연결 층에는 0.5의 드랍률로 드랍아웃을 적용시켰다. 또한 비선형 활성화 함수로는 ReLU를 사용하였고, 학습의 가속화를 위해 Batch Normalization기법을 사용하였다. 또한 제안하는 모델과 비교모델인 LeNet 기반 CNN의 첫 번째 완전 연결 층에는 0.5의 드랍률로 드랍아웃 기법을 적용시켰으며, 두 번째 완전 연결 층은 softmax 계층으로 구성하였다. 학습 방식은 one-hot형태로 encoding된 실제 정답과 모델의 예측 결과 값을 softmax함수로 변형한 뒤 둘을 cross-entropy를 최소화 하는 방식으로 학습하였다. 실험에 대한 학습곡선을 그림 10에 나타내었다. 그림 10에서 NDF는 드랍필터가 적용되지 않았음을 의미하고, DF는 적용된 경우를 의미한다. 또한 Tr과 Te는 각각 학습 셋과 테스트 셋에 대한 결과를 의미한다. LeNet과 Inception은 각각 표 6과 표 7에 제시된 구조를 따르는 비교 모델과 제안 모델을 의미한다. 표 8을 통해 드랍필터와 인셉션 모듈의 효과를 요약하여 나타내었다. 표 8에서 (a) - (b)는 학습 셋에 대한 인식률과 테스트 셋에 대한 인식률의 차이로 과 적합의 정도를 의미한다. 실험을 통해 제안하는 모델이 드랍필터가 적용되지 않은 LeNet 기반 CNN에 비해 약 30%의 학습 변수만을 사용하고도 테스트셋 기준 3.279% 높은 인식 정확도를 보여주었으며, 과 적합의 정도 역시 2.089%로 가장 낮은 정도를 얻을 수 있었다. 또한 주목 할 점은 LeNet 기반 CNN에 드랍필터를 적용할 경우 테스트셋 기준 94.792%의 인식 정확도를 보였다. 이는 비록 드랍필터가 적용되지 않았지만 최신 구조로

여겨지는 인셉션 모듈 기반의 CNN에 상응하는 인식 성능이다.



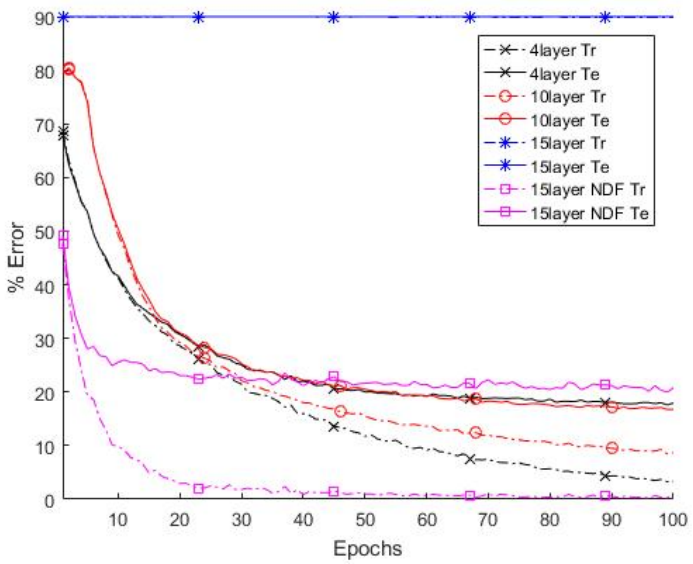
[그림 10] 한글 필기체 인식에 대한 각 모델별 학습 곡선

[표 8] 한글 필기체 인식에 대한 각 모델별 인식 정확도 (학습 좀 더 한 모델로 비교 추가하기)

Model	Accuracy %		
	Training (a)	Test (b)	(a) - (b)
LeNet NDF	99.99	92.581	7.409
LeNet DF	97.919	94.792	3.127
Inception NDF	99.878	94.894	4.984
Inception DF (proposed)	97.949	95.86	2.089

4. 기타 논의사항

각 계층별 드랍필터의 효과를 검증하는 실험 도중 드랍아웃과 드랍필터 모두에 적용되는 문제점을 발견하였다. 학습하고자 하는 모델의 깊이가 매우 깊을 때 드랍아웃 또는 드랍필터를 적용하면 학습이 되지 않는다는 점이다. 일반적으로 알려진 사실은 드랍아웃이 적용된 계층이 늘어날수록 학습속도는 느려진다는 점이다 (Srivastava et. al., 2015). 이번 실험을 통해 모델의 계층이 매우 깊어질 경우 심지어는 학습이 불가능하다는 사실이 드러난 것이다. 그림 11에서 학습곡선을 통해 이러한 현상을 확인할 수 있다. 이 문제에 대한 해결책은 추후 심도 있게 다루어져야 할 내용으로 본 논문에서는 문제를 제기하는 선에서 끝내도록 하겠다.



[그림 11] 드랍필터가 적용된 모델의 깊이별 학습곡선

V. 결 론

본 논문에서는 스마트폰이나 태블릿 PC 등 하드웨어 자원이 제한된 환경에서 필기체 인식 문제를 수행하기 위해 적합한 가벼우면서도 높은 인식 정확도를 보이는 컨볼루션 신경망에 대한 연구를 수행하였다. 이를 위해 인셉션 모듈을 통해 좀 더 가벼우면서도 높은 인식률을 달성할 수 있는 CNN 모델을 제안하였다. 또한, CNN에서 인식 정확도의 극대화를 위해 드랍필터 기법을 제안하는 모델에 적용하였다. 실험 결과 제안하는 모델이 LeNet 기반의 CNN에 비해 약 3배 적은 학습 변수로도 3.279% 높은 인식률을 달성하였다. 또한 드랍필터를 통해 일반화 오류를 낮출 수 있기 때문에 추가적인 글자 클래스에 대해 더욱 적은 학습 데이터로도 높은 인식률을 달성할 수 있을 것으로 기대된다.

하지만 기타 논의사항에서 논의하였듯 드랍필터가 매우 깊은 계층을 가지는 CNN에 적용될 경우 학습이 되지 않는다는 문제점이 있다. 이는 필기체 인식 분야를 넘어 일반적인 이미지 인식 분야에 뛰어난 성능을 보이는 residual network 또는 GoogLeNet에 적용되었을 때 효과적이지 못하다는 것을 의미한다. 이를 해결하기 위한 방법은 추후 연구로 남겨 놓겠다.

추가적으로 고려하고 있는 또 다른 추후 연구로는 실용적은 측면을 더욱 강조하여 영어, 중국어, 일본어, 기호 등을 포함한 훨씬 다양한 범위의 문자를 하나의 모델로 인식할 수 있는 연구를 다룰 예정이다.

참고문헌

Bahlmann, C., Haasdonk, B., & Burkhardt, H. (2002). Online handwriting recognition with support vector machines-a kernel approach. *In Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on* (pp. 49-54). IEEE.

Cho, S. J., & Kim, J. H. (2003, August). Bayesian network modeling of hangul characters for online handwriting recognition. *In Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on* (pp. 207-211). IEEE.

Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint* arXiv:1506.02158.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. *arXiv preprint* arXiv:1302.4389.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint* arXiv:1207.0580.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint* arXiv:1502.03167.

Kang, W. Y., et al. “Hangeul handwriting recognition using deeper convolutional neural networks based on inception modules.” *Korea Computer Congress 2016* (KCC2016), pp. 883-885, 2016.

Kim, I. J., & Xie, X. (2015). Handwritten Hangul recognition using deep convolutional neural networks. I*nternational Journal on Document Analysis and Recognition (IJDAR)*, 18(1), 1-13.

Kim, I. J., Choi, C., & Lee, S. H. (2016). Improving discrimination ability of convolutional neural networks by hybrid learning. I*nternational Journal on Document Analysis and Recognition (IJDAR)*, 19(1), 1-9.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems* (pp. 1097-1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint* arXiv:1312.4400.

Liwicki, M., Graves, A., Bunke, H., & Schmidhuber, J. (2007, September). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. *In Proc. 9th Int. Conf. on Document Analysis and Recognition* (Vol. 1, pp. 367-371).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *In Proceedings of the 27th international conference on machine learning* (ICML-10) (pp. 807-814).

Ng, A. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72(2011), 1-19.

Sermanet, P., Chintala, S., & LeCun, Y. (2012, November). Convolutional neural networks applied to house numbers digit classification. *In Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 3288-3291). IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., & Bregler, C. (2015). Efficient object localization using convolutional networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 648-656).

Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using dropconnect. *In Proceedings of the 30th International Conference on Machine Learning* (ICML-13) (pp. 1058-1066).

Yin, F., Wang, Q. F., Zhang, X. Y., & Liu, C. L. (2013, August). ICDAR 2013 Chinese handwriting recognition competition. *In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on* (pp. 1464-1470). IEEE.

Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint* arXiv:1301.3557.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer International Publishing.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint* arXiv:1611.03530.

ABSTRACT

Sparse Convolutional Neural Network for Handwriting Recognition

Woo-Young Kang

School of Computer Science and Engineering

The Graduate School

Seoul National University

The demands of automatic handwriting recognizer have increased exponentially in various industries such as automatic recognition of postal codes and the number plate of a car. To satisfy these needs, various techniques based on the Convolutional Neural Network(CNN), which has shown remarkable accuracy in many image classification tasks, are applied to handwritten recognition tasks. These studies mainly used deep CNN to achieve high recognition accuracy. However, handwriting recognizer generally used in applications such as smartphone or tablet PC, which is resource constrained environments. Therefore, the memory usage and computational

cost are concurrently considered. In this paper, to reduce model parameters effectively, we applied a CNN model based on the inception module to Hangul Handwriting Recognition(HHR) task. Also, we made our model be sparse using the dropfilter technique to reduce the generalization error and so as to achieve higher recognition accuracy. The inception module is the core component of the GoogLeNet, which won the Imagenet Large Scale Visual Recognition Challenge(ILSVRC) 2014 with 12 times fewer parameters than the winning architecture of ILSVRC 2012. Also, the dropfilter is a modified version of the dropout to get lower generalization error for the CNN architecture. To verify the effect of the dropfilter, we compared the generalization errors of the dropfilter with the dropout using the Canadian Institute for Advanced Research(CIFAR)-10 dataset which consists of 10 classes and 60,000 natural images. Through the verification experiment, we could see that the dropfilter shows lower generalization error than the dropout in a CNN architecture. Also, we found that the effects of the dropfilter in each layer are different. To analyze this, we performed additional experiments to check the difference. After all the verification tasks mentioned above, we conducted HHR tasks using CNN based on the inception module with the dropfilter. The Hangul dataset which was used in HHR tasks consists of 520 classes and 260,000 characters. At the result of HHR tasks, proposed model showed 3.279% higher test set accuracy with 3times fewer parameters than the CNN based on LeNet architecture.

Keywords: Hangul Handwriting Recognition(HHR), Convolutional Neural network(CNN), Dropfilter, Dropout

Student Number: 2015-22892